

**REMARKS**

Claims 1-5 and 25-29 have been cancelled. Claims 6-24 and 30-49 remain in the application. Amended base Claims 6, 30 and 49 have been amended to contain all the limitations upon which those Claims, as originally filed, were dependent. No claim has been allowed.

With regard to the drawings, formal drawings were submitted on January 15, 2004. The drawings were objected to, specifically Fig. 1 and Fig. 2, as being larger than the allotted space and therefore not fitting on the page. After careful examination of Fig. 1 and Fig. 2, Applicant must respectfully disagree with the Examiner's objection to the above-referenced figures because they are in compliance with the page size, margin and sight requirements of 37 C.F.R. § 1.84. For convenience of the Examiner, the replacement (formal) drawings are now being re-submitted with this amendment. Attached are 19 sheets showing Figs. 1-19 for filing in the subject patent application. No new matter is being introduced. Acceptance is respectfully requested. If the Examiner is still of the opinion that the submitted drawings do not comply with 37 C.F.R. 1.84, clarification of the specific error in the drawing(s) is requested.

**Claim Rejections Under 35 U.S.C. § 101**

Claims 1-49 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Claims 1-5 and 25-29 have been cancelled. Applicant respectfully requests reconsideration of Claims 6-24 and 30-49, as amended.

The Examiner states that the claimed computer-related processes of the present invention do not meet the criteria for a statutory process because they do not "result in a physical transformation outside the computer for which a practical application" is either disclosed in the specification or would have been known to a skilled artisan, or are "not limited to a practical application." Applicants must respectfully disagree. The claims as originally filed are indeed limited to a practical application, namely efficient transaction processing in Multi-Version Data Base Management Systems, thereby eliminating delays and aborts for resolving read-write database conflicts.

Section 2106(II)(A) of the MPEP notes that in order to establish a prima facie case of non-statutory subject matter, the Examiner must show that the claimed invention as a whole (1) is directed solely to an abstract idea or to the manipulation of an abstract idea or (2) does not produce a useful result. The MPEP further notes that only when the claim is devoid of any limitation to practical application in the technical arts should it be rejected under 35 U.S.C. § 101. Further, when such a rejection is made, the Examiner must expressly state how the language of the claims has been interpreted to support the rejection. Applicant believes that the Examiner has failed to meet this standard.

The applicant is in the best position to explain why an invention is believed useful. The Examiner should therefore focus her efforts on pointing out statements made in the specification that identify all practical applications for the invention. The Examiner should rely on such statements throughout the examination when assessing the invention for compliance with all statutory criteria.

Nonetheless, Applicant has amended Claims 6-24 and 30-49 to expedite prosecution. Applicant believes that the amended claims are in condition for allowance.

#### **Claim Rejections Under 35 U.S.C. § 103**

Claims 1-49 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bamford, *et al.* (U.S. Patent 6,237,001) in view of “Efficient and Flexible Methods of Transient Versioning of Records to Avoid Locking by Read-Only Transactions” by Mohan, *et al.* Claims 1-5 and 25-29 have been cancelled.

Amended Claims 6-24 and 30-49 are directed to a multi-version database system and method to control visibility of data during transaction processing. A transaction includes a transaction identifier that identifies the transaction and an invisibility list of transactions whose effects are invisible to the transaction. The purpose of the invisibility list is to encapsulate exceptions to the basic rule of visibility, namely that a transaction can, by default, “see” the

changes produced by other transactions with transaction IDs that are less than or equal to its own transaction ID, but cannot “see” those changes produced by transactions with transaction IDs that are greater than its own transaction ID.

Changes made by other transactions are visible to the transaction based on the isolation level of the transaction. Records are visible to a transaction based on a creator identifier stored in the record that identifies the transaction that created the record. The ANSI SQL92 standard defines four levels of transaction isolation: Read Uncommitted, Read Committed, Repeatable Read and Serializable. These four levels of transaction isolation result in different possible outcomes for the same transaction scenario. That is, the same work performed in the same fashion with the same inputs may result in different answers, depending on the isolation level. These levels are defined with the classical serializability definition, plus three phenomena or anomalies that are either permitted or not at a given isolation level: Dirty Read, Non-repeatable Read and Phantom.

Isolation Level	Dirty Read	Nonrepeatable Read	Phantom Read
READ UNCOMMITTED	Permitted	Permitted	Permitted
READ COMMITTED	Not Permitted	Permitted	Permitted
REPEATABLE READ	Not Permitted	Not Permitted	Permitted
SERIALIZABLE	Not Permitted	Not Permitted	Not Permitted

The preferred embodiment of the present invention supports the first three isolation levels: Read Uncommitted, Read Committed and Repeatable Read. In addition it can support the visibility aspects of Serializable isolation (the safest isolation level) in avoiding dirty read, non-repeatable read and phantom anomalies.

In contrast, Bamford, *et al.* relates to a method and apparatus for managing access to data on a distributed database system. A “snapshot list” is generated for a transaction executing on the distributed database system. The snapshot list specifies snapshot times for a plurality of locations in the distributed database system. The snapshot times are determined based upon the

location of a data item and the location where the transaction is executing. The selection of a version of the data item to be provided to the transaction is made based upon the snapshot time for the location associated with the data.

Snapshot Isolation, as applied in Bamford, *et al.*, is indeed a type of a multiversion concurrency control algorithm and provides an isolation level that avoids many of the common concurrency anomalies. Snapshot Isolation guarantees that all reads made in a transaction will see a consistent snapshot of the database, and the transaction, itself, will successfully commit only if no updates it has made conflict with any concurrent updates made since the snapshot.

A transaction  $T_i$  executing under Snapshot Isolation conceptually reads data from the committed state of the database as of time  $\text{start}(T_i)$  (the snapshot), and holds the results of its own writes in local memory store, so if it reads data it has written it will read its own output. Predicates evaluated by  $T_i$  are also based on rows and index entry versions from the committed state of the database at time  $\text{start}(T_i)$ , adjusted to take  $T_i$ 's own writes into account.

The interval in time from the start to the commit of a transaction is called its transactional lifetime. Two transactions are concurrent if their transactional lifetimes overlap. Writes by transactions active after  $T_i$  starts (i.e., writes by concurrent transactions) are not visible to  $T_i$ . When  $T_i$  is ready to commit, it obeys the First Committer Wins rule, as follows:  $T_i$  will successfully commit if and only if no concurrent transaction  $T_k$  has already committed writes of rows or index entries that  $T_i$  intends to write. Reading from a snapshot means that a transaction never sees the partial results of other transactions:  $T$  sees all the changes made by transactions that commit before  $\text{start}(T)$ , and it sees no changes made by transactions that commit after  $\text{start}(T)$ .

The ANSI SQL92 standard relied upon in the present invention differs greatly from the Snapshot Isolation method used in Bamford, *et al.* "A Critique of ANSI SQL Isolation Levels" by Berenson, *et al.* cites Snapshot Isolation as an example of an isolation level that does not exhibit the standard anomalies described in the ANSI SQL92 standard. The ANSI SQL92

isolation level Repeatable Read, as supported by the present invention, allows phantom reads, but prevents non-repeatable reads. In contrast, Snapshot Isolation, as applied in Bamford, *et al.*, allows non-repeatable reads, but prevents phantom reads.

With regard to the Examiner's rejection of Claim 1, Claim 1 has been cancelled. However, all the limitations of Original Claims 1-5 have been incorporated into Amended base Claim 6. There is no suggestion in Bamford, *et al.* at column 8, line 20 to column 9, line 5 of an invisibility list which identifies other transactions whose effects are to be invisible to the requesting transaction.

According to Applicant's Claim 6, as now amended, the invisibility list contains a vector of transaction IDs, along with the total number of transaction IDs in the vector, and the number of transaction IDs in the vector that should be treated as invisible. Next, the vector of transaction IDs contains the transaction IDs of other transactions that are to be invisible to the transaction, even though their transaction IDs are ordered before the transaction ID of the transaction. Lastly, the vector of transaction IDs contains a list of transaction IDs of other transactions that are to be visible to the transaction (a visibility list), even though their transaction IDs are ordered after the transaction ID of the transaction. The vector of transaction IDs contains no more than 32 transaction IDs.

In contrast to this, the method in Bamford, *et al.* includes the step of generating a MUST SEE time and a CANNOT SEE time for each of the plurality of locations. In Bamford, *et al.*, two transaction sets are maintained for each serializable transaction, including a MUST-SEE set and a CANNOT-SEE set. The term set as used in the definitions of the MUST-SEE set and CANNOT-SEE set implies that the transactions are divided into two groups: those that made updates that have been seen by the serializable transactions and those that made updates that were removed from data that has been seen by the serializable transaction and all transactions that subsequently updated the seen data.

The MUST-SEE set and CANNOT-SEE set of Bamford, *et al.* are different and distinct from the invisibility list in the present invention. Bamford, *et al.* makes no reference to an invisibility list that contains a vector of transaction IDs, along with the total number of transaction IDs in the vector, and the number of transaction IDs in the vector that should be treated as invisible, the transaction IDs of other transactions that are to be invisible to the transaction even though their transaction IDs are ordered before the transaction's ID, and a list of transaction IDs of other transactions that are to be visible to the transaction. In the present invention, the visibility list is contained within the vector of transaction IDs that comprises the invisibility list.

Thus, not only does Applicant's invention require an invisibility list that identifies other transactions whose effects are to be invisible to the transaction, but also requires that the invisibility list contain a vector of transaction IDs, the total number of transaction IDs in the vector, and the number of transaction IDs in the vector that should be treated as invisible, the transaction IDs of other transactions that are to be invisible to the transaction, and a list of transaction IDs of other transactions that are to be visible to the transaction. Bamford, *et al.* does not have an invisibility list, never mind the specifically claimed vector of transaction identifiers by Applicants.

Further, there is also no suggestion in Bamford, *et al.* at column 5, lines 25-41 of ANSI SQL92 isolation levels which describes whether changes made by other transactions are to be visible to the transaction. Bamford, *et al.* makes no mention supporting three of the four ANSI SQL92 levels of transaction isolation. Bamford, *et al.* only supports Snapshot Isolation and does not have any concept of specifying an "isolation level" parameter as part of a transaction ID. This factor is also present in Amended Claim 6.

With respect to the elements of previous Claim 4, now incorporated into Claim 6, Bamford, *et al.* at column 2, lines 22-30 in combination with Mohan, *et al.* fails to disclose transaction identifiers associated with transactions that operate in the present having an even numeric value, and transaction identifiers associated with transactions operating "as-of" a

determined time in the past having an odd numeric value. Bamford, *et al.* makes no use of an “as-of” time.

With respect to the elements of previous Claim 5, now incorporated into Claim 6, Bamford, *et al.* at column 8, line 20 to column 9, line 5 in combination with Mohan, *et al.* fails to disclose finding a transaction identifier for an earliest transaction that started on or after the specified “as-of” time; creating a new transaction, the new transaction having a start-time equal to the specified “as-of” time; a transaction identifier equal to the transaction identifier for the earliest transaction, less one; and an isolation level set to Read Committed; or initializing the invisibility list of the new transaction to include the transaction identifiers of all transactions having transaction identifiers values less than the transaction identifier for the new transaction and end-times greater than the specified “as-of” time. Bamford, *et al.* makes use of neither an “as-of” time, the ANSI SQL92 Read Committed level nor invisibility lists.

With respect to Claim 7, Bamford, *et al.* at column 8, line 20 to column 9, line 5 in combination with Mohan, *et al.* at section 2, lines 28-30 and Fig. 3 makes no reference to the requesting transaction further comprising an associated visibility list. The claimed visibility list is a list of transaction IDs of other transactions that are to be visible to the transaction and is contained in a vector of TIDs. The MUST-SEE set of Bamford, *et al.* includes all of the transactions that made updates that have been seen by the serializable transaction. Bamford, *et al.* makes no use of a vector of TIDs let alone a list of transaction IDs.

Claim 11 has now also been amended and rewritten in independent form and is believed to be patentable over the cited prior art. Claim 11 is directed to a method for rolling back the changes made by a transaction by storing an “aborted transaction identifier” in a deleter transaction ID field, a deletion descriptor, and a NULL transaction identifier.

The Examiner had previously refused to consider those additional features to distinguish over the prior art because of the optional “if present” language in the original claim. As this language is now deleted from Claim 11, reconsideration is requested.

Similarly, Claim 17 has also been rewritten to remove the optional “if present” restriction on the deleter transaction identifier and deleted record(s). Claim 17 should now be considered in full. It is patentable over Bamford, *et al.* and the other prior art, which do not consider Repeatable Read isolation levels or deleted transactions.

With regard to Amended Claims 30 and 49, the system claim and means-plus function claim corresponding to the Amended method Claim 6, respectively, the same arguments above as applied to Amended Claim 6 apply. With regard to the Examiner’s rejection of the remaining Claims, the same arguments above apply.

With respect to Claims 6-24 and 30-49, the Examiner has failed to meet the burden of persuasion for a prima facie case of obviousness. Mohan, *et al.* does not overcome any of the deficiencies of Bamford, *et al.*

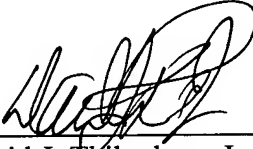


**CONCLUSION**

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By   
David J. Thibodeau, Jr.  
Registration No. 31,671  
Telephone: (978) 341-0036  
Facsimile: (978) 341-0136

Concord, MA 01742-9133

Dated:

10/19/06